OCP's Rack Manager Controller subproject (OpenRMC)

John Leung, Principle Engineer
Intel Corporation

Han Wang, Senior Architect
Inspur
OpenRMC Overview

OpenRMC

• Goals and Motivation
• Logistics and meetings

Status

• Reviewed of existing rack management implementations
• Specifying interface and requirements
• Received Code Contributions

RMC for openEdge Platform
The OpenRMC Goals

Specify the Rack Manager Controller service interfaces

• Northbound interface to datacenter manager (spec)
• Southbound interface requirements to OCP platforms in the rack

Deliver a Rack Manager implementation

• Available as open source

OCP compliant hardware designs

• Handled by other OCP projects
Motivation for OpenRMC

1. System Firmware (BIOS)
   - OCP System Firmware project

2. BMC Firmware
   - OpenBMC governed by Linux Foundation

3. Rack Manager Software/Firmware
   - OCP OpenRMC

- With OpenBMC, the industry unified the various repositories in 2018
- With rack manager, OCP will provide a source repository and prevent splintering
The RMC can be hosted in various locations

**OpenRACK**  
(within power shelf)

**EIA, OpenRACK**  
(within switch)

**Olympus**  
(standalone)
Logistics

• A subproject of the Hardware Management project
  • John Leung (Intel) and Han Wang (Inspur) are co-chairs
  • Wiki
  • Mail-list

• Participation
  • ARM, Microsoft, Facebook, Huawei, Inspur, Nokia, Intel, etc
OpenRMC project has since Nov 2018

Reviewed and compare

• Existing RMC interfaces and architectures from Facebook, Intel, Inspur and Microsoft (Comparison of interfaces)

Draft specifications and requirements

• Draft of Northbound API Specification
• Draft of Southbound interface requirements

Reviewed RMC code contributions

• From Microsoft, Inspur, and Intel
## Source Repository

On OCP Github (github.com/opencomputeproject/Rack-Manager)

Contrib-Microsoft, Contrib-Inspur, Contrib-Intel folders

./OpenRMC folder

<table>
<thead>
<tr>
<th>Language</th>
<th>Architecture</th>
<th>Processor</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft</td>
<td>C++</td>
<td>OpenBMC</td>
<td>AST2500</td>
</tr>
<tr>
<td>Inspur</td>
<td>C++</td>
<td>OpenBMC</td>
<td>AST2500</td>
</tr>
<tr>
<td>Intel</td>
<td>C++</td>
<td></td>
<td>Intel NUC</td>
</tr>
</tbody>
</table>
OpenRMC project goals

Replacing firmware binaries with buildable source
• An OCP location exists for RMC sources
• OCP RMC contribution can reference the source repository

Open RMC
• RMC source that conform to the OpenRMC Northbound API
Nokia Airframe openEdge Chassis

Open edge chassis overview

Key specifications

• Cooling: Fan units are part of sled solution
  • Air flow direction configurable: front to rear/rear to front

• Chassis management controller (RMC)
  • PSU management (control, sensors, ..)
  • Management Ethernet interface to sleds
    • 1 GE to all sleds via backplane
    • 1x 1 GE (RJ45) + 2x 10 GE (SFP+) front panel interface for external connectivity and chaining of multiple chassis

• Power distribution board and chassis backplane provide connectivity between RMC, sleds and PDUs
Airframe openEdge RMC

RMC
Management unit

Chassis management controller (RMC)
- PSU management (control, sensors, ..)
  - Control and supervision of PSUs
  - Access to sensor data (voltages, currents, power consumption)
- RMC controller from AST2500 family
  - USB debug port in front panel
- On-board unmanaged Ethernet switch simplifies HW management connectivity
  - Single management interface for entire chassis
    - 1 GE management Ethernet interface to all sleds via backplane (1000BASE-T)
    - 1x 1 GE (RJ45, 1000BASE-T)
    - 2x 10 GE (SFP+) front panel interface for external connectivity and chaining of multiple chassis

Interface supported:
- DMTF Redfish
- DMTF SMASH
- IPMI
- SNMP
Testing OpenRMC conformance

- The **OpenRMC profile** file represents the Northbound interface specification in a JSON format (Redfish resources & properties)
- The Redfish Interop Validator: 1) reads the OpenRMC profile, 2) autogenerates the tests, and 3) runs them against an implementation
- The Interop Validator is a component of the Redfish conformance test suite, which also includes
  - Redfish Service Conformance Check
  - Redfish Service Validator
  - Redfish Usecase Checkers
Status against Goals

Specify the Rack Manager Controller service interfaces
- OpenRMC Northbound interface (v0.3) in review
- Expect to be proper subset of openEdge RMC requirements

Deliver a Rack Manager implementation
- Three RMC code repositories in an OCP Github location

OCP compliant hardware designs
- Expect contributions to conform to the OCP OpenRMC profile
- Expect contributions to include buildable source, instead of a firmware image
Call to Action

- Join the OpenRMC discussion
  - OpenRMC@OCP-All.groups.io
- Participate in the OpenRMC meetings
  - opencompute.org/wiki/Hardware_Management/Open_RMC
- Contribute to the OpenRMC source & profile
  - github.com/opencomputeproject/Rack-Manager
  - github.com/opencomputeproject/OCP-Profiles
Open. Together.

OCP Regional Summit
26–27, September, 2019